REMARKS

Reconsideration and allowance of the above-referenced application are respectfully requested. Claims 9, 11 and 12 are amended, new claim 21 is added, and claims 1-3 and 5-21 are pending in the application.

Claims 1 and 5-8 stand rejected under §103 in view of U.S. Patent No. 5,949,786 to Bellenger in view of Newton. This rejection is respectfully traversed.

Bellenger neither discloses nor suggests combining the first and second hash keys (generated from respective first and second layer 3 information), as specified in claim 1 (and claims 11 and 16). Rather, Bellenger teaches in Fig. 4 that a certain fields are selected for input to a hash generator 414, resulting in a single hash key (col. 5, lines 52-57). In addition, Fig. 5 shows that only one hash value is selected (based on a flow select 501) for accessing a route table (col. 6, lines 21-32).

The Official Action applies a tortured interpretation of Bellenger that is inconsistent with the teachings of the reference. In particular, the Examiner asserts in paragraph 7 that:

> Bellenger discloses ... generating first and second hash codes **(specific bytes from a particular field)** according to a prescribed hash function in response to first and second layer 3 information (IP source and IP destination) within the received data packet, respectively (col. 3, lines 9-32; col. 4, lines 40-61; col. 5, line 17-col. 6, line 59, esp. col. 6, lines 34-49; and col. 8, lines 4-36); combining the first and second hash codes according to a prescribed combination into a signature (hash value) for the received data packet (col. 3, lines 9-32; col. 4, lines 40-61; col. 5, line 17-col. 6, line 59, esp. col. 6, lines 34-49; and col. 8, lines 4-36)....
>
> (Emphasis added).

The Examiner also asserts that "**each hash code is a hash key**". However, the reliance on Newton is misplaced because the definition relied on is a "key", and not a **hash key**, as claimed. Moreover, the Examiner appears to take the unreasonable position that any byte from a particular

field can be construed as a “hash code”; as shown below, however, a hash key or hash code **must** have been generated **according to a prescribed hashing operation.**

Bellenger **actually discloses** that only a single hash key is generated from selected portions of the received data frame. For example, Bellenger describes that Figure 4 illustrates a received data frame 400, where the flow detect logic retrieves data from selected fields in order to generate a **single** pseudo-random tag:

> The flow detect logic runs a pseudo-random hash algorithm over selected fields in the control header of the frame to generate a pseudo-random tag. Thus, the field 410, the field 411, the field 412, and the field 413 are selected for input into a hash generator 414. The tag generated by the hash generator 414 is supplied on line 415 for use in accessing the route table 416.
>
> (Col. 5, lines 53-59).

In addition, Bellenger describes with respect to Figure 5 that multiple hash flow analyses may be executed in parallel in order to discern between multiple protocols (e.g., IP routing, AppleTalk, etc.), and that a multiplexer 502 is used to select only one of the N flows for output onto the line 503 to be used as a hash value for accessing the route table by the CPU:

> The flow detect logic in a preferred system executes a plurality of hash flow analyses in parallel as illustrated by FIG. 5. Thus in FIG. 5, a received frame is supplied on line 500 in parallel to hash flow logic 1 through hash flow logic N, each flow corresponding to a particular frame format. Also, the received frame is supplied to a hash flow "select" 501 which is used for selecting one of the N flows. The output of flows 1 through N are supplied through multiplexer 502 in FIG. 5, which is controlled by the output of the select flow 501. The output of the select flow 501 causes selection of a single flow on line 503, which is used as a hash value for accessing the route table by the CPU.
>
> (Col. 6, lines 22-33).

Moreover, a review of Figure 6 and the associated description in columns 7-8 demonstrates that Bellenger provides <u>no disclosure whatsoever</u> of generating a packet signature by generating first and second hash keys, let alone combining the first and second hash keys into a signature for the received the packet.

> FIG. 6 illustrates a preferred hardware architecture for implementing the flow detect filters and the route table accessing logic according to the present invention. Thus, flow detect logic includes a plurality of flow detect filters. Each filter includes a template register 600 which specifies <u>bytes of an incoming frame which are to be used as a hash seed for the generation of the hash code</u>. The template stored in register 600 specifies all protocol dependent fields for a particular protocol. The fields are not distinguished in the template register 600 beyond indicating whether they are included as a seed for the hash or not. As the frame is processed, each byte in the initial header of the frame is either included in the hash function seed or it is ignored.
>
> (Col. 7, lines 12-22).
>
> The mask logic 617 receives the template from the template register 600 on line 618, and the incoming frame on line 610. <u>A pseudo-random seed value 619 is generated by the mask 617 for supply to a pseudo-random number generator 620. The pseudo-random number generator 620 supplies a hash code on line 621 to the hash result register 604</u>. The output of the filter is supplied on line 624 from the result register 604 and hierarchy register 602 to priority logic 625. Other inputs to the priority logic 625 are indicated on lines 626. <u>Each of these lines 626 is coupled to a filter similar to that coupled to line 624</u>.
>
> <u>The selected hash value</u> is provided on line 630 as <u>a hash value 631</u> used for accessing the route table 632.

Hence, Bellenger describes that the template 600 (specifying the bytes to be used in generating the hash seed) is supplied to the mask, which outputs the <u>selected bytes</u> in the form of a "seed value 619" to the pseudo-random number generator 620 in order to generate **the single hash code** output on line 621 and 624 for the **corresponding protocol**.

Hence, Bellenger merely describes that: (1) a hash code is generated based on multiple protocol-dependent values from a data frame for a corresponding prescribed protocol; and (2) multiple hash codes may be generated for **respective** protocols, where a multiplexer circuit 625 outputs a **selected hash value** to determine whether a "hit" is detected by a comparator 635 (see col. 7, line 67 to col. 8, line 3).

As described above, Bellenger provides <u>no disclosure whatsoever</u> of generating <u>first and second hash keys</u> to generate <u>a single packet signature</u>, as claimed. The Examiner's assertion that the first and second hash codes are taught by "specific bytes from a particular field" is both nonsensical and <u>inconsistent with the explicit teachings of the reference</u>. Rather, Bellenger explicitly describes that only a <u>single</u> hash code is generated, and that a pseudo-random number generator 620 outputs the hash code onto line 621 based on the selected portions of the packet from the mask 617.

Further, column 6, lines 34-49 (cited by the Examiner as teaching "combining hash codes from a plurality of fields [sic]"<u>actually</u> summarizes the above-quoted description of col. 6, lines 22-33 with respect to Figure 5:

> Thus a preferred embodiment of the present invention uses a switching technique base [sic] on flow signatures. Individual frames of data move from one of the Ethernet ports to a shared buffer memory at the node. As the data is being moved from the input port to the buffer, a series of hash codes is computed for various sections of the input data stream. Which bits are or are not included in each hash calculation is determined by a stored vector in a vector register corresponding to that calculation. For example, in the most common case of an IP frame, the hash function starts at the 96th bit to find the "0800" code following the link-layer source address, it then includes the "45" code, 32 bits of IP source, 32 bits of IP destination, skips to protocol ID 8 bits, and then at byte 20 takes the source port 16 bits and the destination port 16 bits. The result is a 64 bit random number identifying this particular IP flow.

(Col. 6, lines 33-49).

There is <u>no</u> disclosure or suggestion of generating a packet signature based on <u>first and second hash keys</u>, as claimed.

For these and other reasons, the rejection of claims 1 and 5-8 should be withdrawn.

Claim 11 stands rejected in view of Bellenger and U.S. Patent No. 5,757,795 to Schnell. This rejection is respectfully traversed. The comments above with respect to Bellenger are incorporated in their entirety herein by reference.

As admitted in the Official Action, Bellenger does not disclose generating a packet signature by a network switch port of the integrated network switch for a data packet received at the network switch port based on performing the prescribed has operation on the first and second portions of the layer 3 information and a corresponding received in a packet.

Schnell merely discloses that each network port 104 includes a hash logic 306 for reducing the bit size of each layer 2 source/destination address (e.g., MAC address) to a smaller value, and for providing the hashed result through a hash bus interface 308 and a hash bus 218 to a hash memory 217. (See, e.g., col. 7, lines 42-45; col. 8, lines 19-26 and 51-60).

However, Schnell provides no disclosure or suggestion of performing hashing of layer 3 information, as claimed. Moreover, there is no disclosure or suggestion that the teachings of Schnell could be modified to apply to selected portions of layer 3 address information, as claimed. "The mere fact that the prior art may be modified in the manner suggested by the Examiner does not make the modification obvious unless the prior art suggested the desirability

of the modification." In re Fritch, 23 USPQ2d 1780, 1783-84 (Fed. Cir. 1992). In re Mills, 16 USPQ2d 1430 (Fed. Cir. 1990).

Regardless, the hypothetical combination still would neither disclose nor suggest the claimed feature of generating first and second hash keys, for the first and second portions of the corresponding layer 3 information in the layer 3 switching entry, and combining the first and second hash keys to form the entry signature.

Hence, the rejection of claim 11 as amended should be withdrawn.

Claims 16-19 stand rejected under §103 in view of Bellenger, Newton, U.S. Patent No. 5,555,405 to Griesmer et al., and Schnell. This rejection is respectfully traversed. The comments above with respect to claims 1 and 11 are incorporated in their entirety herein by reference.

As argued above, Bellenger plus Newton neither discloses nor suggests generating a packet signature by generating first and second hash keys for the first and second portions from the data packet bsed on a prescribed hash operation. Rather, Bellenger uses different portions of the packet to generate a single hash key. There is no disclosure or suggestion of generating a packet signature based on first and second hash keys, as claimed.

Further, Griesmer et al. merely describes a conventional hash table used as an index based on each hash table entry containing a pointer to a forwarding entry set within the forwarding table. However, Griesmer et al. neither discloses nor suggests, singly or in combination with Bellenger, Newton, or Schnell, generating a **packet signature** by generating **first and second hash keys**, as claimed.

In fact none of the applied references, singly or in combination, disclose or suggest the claimed feature of generating a **packet signature** by generating **first and second hash keys**, as claimed.

For these and other reasons, the rejection of claims 16-19 should be withdrawn.

It is believed the dependent claims are allowable in view of their dependency from their respective independent claims.

In view of the above, it is believed this application is in condition for allowance, and such a Notice is respectfully solicited.

To the extent necessary, Applicant petitions for an extension of time under 37 C.F.R. 1.136. Please charge any shortage in fees due in connection with the filing of this paper, including any missing or insufficient fees under 37 C.F.R. 1.17(a) or 1.17(e), to Deposit Account No. 50-0687, under Order No. 95-333, and please credit any excess fees to such deposit account.

Respectfully submitted,

Leon R. Turkevich
Registration No. 34,035

Customer No. 20736
Date: September 20, 2004